## REMARKS

Claims 1-40 are pending in the present application. No claims were canceled; claims 5, 8, 10, 14, 18, 27, and 35 were amended; and claims 41 - 45 were added. Reconsideration of the claims is respectfully requested.

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

In response to the objection to the drawings, formal drawings will be filed subsequently.

### I. 35 U.S.C. § 112, Second Paragraph

The examiner has rejected claims 35-40 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. This rejection is respectfully traversed.

Claim 18 clearly recites that the "medium" is computer usable, but more importantly, the claim further defines the relationship between the computer and the medium by reciting, "the instructions are embodied within the computer usable medium." One of ordinary skill in the art would easily understand the term "computer usable medium" as a medium utilized by a computer. Moreover, since instructions are embodied on the medium, no other conclusion is possible.

Therefore the rejection of claims 35-40 under 35 U.S.C. § 112, second paragraph has been overcome.

### II. 35 U.S.C. § 102, Anticipation

The examiner has rejected claims 1, 8, 10, 11, 13, 18, 26, 30, and 35 under 35 U.S.C. § 102 as being anticipated by *Beckman et al.* (U.S. Patent No. 4,340,965; hereinafter referred to as "*Beckman*") This rejection is respectfully traversed.

The present invention relates to a mimic device for mimicking or emulating a device that is not responding. In accordance with a preferred embodiment of the present invention, the mimic device has no advance indication whether or not a device is present and operational on a bus. Therefore, the mimic device must emulate a device only when

the device itself does not respond. This is accomplished by detecting a signal on the bus indicating a request to access the device. Once such a signal to the device has been detected the mimic device monitors the bus for a response by the device. Again, the mimic device has no advance information as to whether the device is on the bus and operational or not. Therefore the mimic device must monitor the bus for a response from the device for the detected signal. Of course, the device will nether respond that it is not present or respond that it is not operational, so the mimic device will only send a response to the signal when a selected period of time passes without a response being made by the device.

Claim 1 recites:

> A method in a data processing system for mimicking a device for use within the data processing system, wherein the device may be connected to a bus, the method comprising:
>
> detecting a signal on the bus indicating a request to access the device;
>
> monitoring the bus for a response by the device; and
>
> sending a response to the signal when a selected period of time passes without a response being made by the device.
>
> (Similar limitations are recited in each of the base claims)

The examiner asserts that Beckman teaches each of the above limitations. A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). Beckman does not teach or suggest the present invention as recited in the claims.

Beckman teaches to prevent a single malfunctioning device from interrupting the communications of the remainder of the system by monitoring signals originating with the suspect device, not from elsewhere as required by claim 1 and all other independent claims. Beckman teaches:

> In essence, the subject invention utilizes a timing mechanism that is enabled to measure a predetermined period of time when an electronic device associated with the line being monitored by the subject invention transmits a low signal, such as the start bit, after a high signal. Such timer is reset by a high signal on the line, and thus may be reset by either a data bit that is high or the stop bit at the end of the character. The timing of the predetermined period of time is started each time a low signal is transmitted after a high signal. If the end of the predetermined time period is reached and the

electronic device associated with the line is not providing a high signal, the subject invention assumes that a malfunction has occurred and forces the output to a high state, thus simulating an idle line. Under normal conditions, the stop bit would reset the timer before the predetermined period of time expires, thereby allowing communication to be carried on normally. If desired, but not preferred, the timing mechanism may be reset only by the stop bit at the end of the character.
(col. 3, lines 3 - 23)

Beckman does not teach "**detecting a signal on the bus indicating a request to access the device**", but instead teaches detecting a start bit from the device itself. Beckman does not teach or suggest to detect a signal on a bus indicating a request to access the device. The signal on the line indicating a request to access the device may have long since been sent at this point. In fact, Beckman's teaching does not rely on any request to access the device whatsoever. Instead, once a start bit is detected, the timer is initiated. There is no reliance on any indication of a request for starting the process. In fact, there need not be a request to access the device, according to Beckman. The device merely transmits a start bit, and if a stop bit is not received during a preset time period, the device is assumed to be faulty and the data line is forced high.

The present invention, on the other hand, functions regardless of whether or not the device is actually on the bus. Once a signal is detected on the bus indicating a request to access the device, the present invention monitors the bus for a response from the device. In contrast, Beckman teaches that the devices itself must transmit the first bit, or 'start bit'. If a start bit does not occur, Beckman's invention will not force the signal line high (mimic the device). In fact, there is no need to respond since the line was never initially forced low by the device.

Beckman discloses a completely different scheme, from that of the present invention, which is not applicable to problems being solved by the present invention.

Furthermore, *Beckman* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *Beckman* actually teaches away from the presently claimed invention because it teaches responding to a signal **from** the device being mimicked opposed to a responding to a signal **to** the device being mimicked as in the presently claimed invention. Absent, the examiner pointing out some teaching or inventive to implement *Beckman* and responding to a signal to the device being

mimicked one of ordinary skill in the art would not be led to modify *Beckman* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Beckman* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

Since claims 2-13, 19 - 26, 31 - 34 and 36 - 40 depend from claims 1, 18, 30, and 35, respectively, the same distinctions between *Beckman* and the claimed invention in claim 1, 18, 30, and 35 for these claims. Additionally, claims 2-13, 19 - 26, 31 - 34 and 36 - 40 claim other additional combinations of features not suggested by the reference. Consequently, it is respectfully urged that the rejection of claims 2-13, 19 - 26, 31 - 34 and 36 - 40 have been overcome.

Therefore, the rejection of claims 1, 8, 10, 11, 13, 18, 26, 30, and 35 under 35 U.S.C. § 102 has been overcome.

The examiner has rejected claims 1-26 and 35 under 35 U.S.C. § 102(b) as being anticipated by *Windows 95* ® operating system. (hereinafter referred to as "*Win95*" .) This rejection is respectfully traversed.

Initially it is noted that the examiner has averred a statement of personal knowledge which is unverifiable and does not comply with MPEP 707.05. It is therefore respectfully requested that the examiner either provide a written reference describing the facts as asserted by the examiner, or, submit an examiner's affidavit with all relevant knowledge as understood by the examiner in compliance with 37 CFR 1.104(d)(2). Furthermore, since this rejection is improper in its present form, it is asserted that an non-final office action should be issued.

However, in an effort to expedite the prosecution on the present application, Applicant will distinguish the present invention from prior art 'plug-and-play' methods. Contrary to assertion by the examiner, a plug and play system initially scans the entire system for new hardware every time the system is booted or during Power On Self Test (POST). POST is merely a series of built-in diagnostics that are performed during start up. The plug-and-play technology performs a variety of different tasks commencing with scanning for legacy devices, or non plug-and-play devices. Legacy devices cannot change their configuration settings like plug-and-play devices so the legacy devices must

be accommodated prior to configuring the plug-and-play devices. If a plug-and-play device is not present at start up, it is not registered and configured.

Nothing in the plug-and-play environment teaches or suggests "**detecting a signal on the bus indicating a request to access the device**", "**monitoring the bus for a response by the device**" and "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the present claims. Plug-and-play merely provides a convenient way for a user to add new device to a plug-and-play compatible system. Plug-and-play automatically configures the new devices around existing, non-configurable, legacy devices. If a device is not present in the POST sequence, it will not be registered. If that missing device is needed for system start- up, a system error is generated and system start-up is halted.

In contrast with plug-and-play, the present invention detects a signal on the bus that is intended for a device, whether the device is present, absent, operational, or faulty. Plug-and-play merely polls existing device during the POST self test. Plug-and-play does not detect any signal, it works with the POST self test for **identifying** devices which are present. If a response is not received for a device, the device is simply not configured in the system. Conversely, the mimic device in the present invention responds or mimics a non-responsive device. No system error is generated, and by utilizing the present invention a system can complete system start-up with a missing or defection device, even if that device must respond during POST. The mimic device provides the necessary response subsequent to the expiration of the time period.

Since claims 2 - 13, 15 - 17, 19 - 26 and 36 - 40 depend from claim 1, 14, 18 and 35, respectively, the same distinctions between *Win95* and the claimed invention in claim 1, 14, 18 and 35, for these claims. Additionally, claims 2 - 13, 15 - 17, 19 - 26 and 36 - 40 claim other additional combinations of features not suggested by the reference. Consequently, it is respectfully urged that the rejection of claims 1-26 and 35 have been overcome.

Therefore, the rejection of claims 1-26 and 35 under 35 U.S.C. § 102(b) has been overcome.

## III. 35 U.S.C. § 103, Obviousness

The examiner has rejected claims 1-40 under 35 U.S.C. § 103(a) as being unpatentable over any one of *Blackborow et al.* (U.S. Patent No. 5,297,067; hereinafter referred to as "*Blackborow*"), *Emerson et al.* (U.S. Patent No. 5,898,861; hereinafter referred to as "*Emerson*"), and *Hartung et al.* (U.S. Patent No. 5,920,709; hereinafter referred to as "*Hartung*"). This rejection is respectfully traversed.

The present invention, as embodied in the claims, recite mimicking or emulating a device that is not responsive. The way in which the present invention determines if a device is responsive is by monitoring a bus for a selected period of time for a response. In accordance with the present invention, **the mimic device has no advance information as to whether the device is on the bus and operational or not**. Therefore the mimic device must monitor the bus for a response from the device for the detected access request signal. This is simply not taught or suggested by any reference or combination of references cited by the examiner.

According to each reference cited by the examiner where a device is mimicked, the mimicking device knows in advance that the device to be mimicked is not available. In no case does the 'mimicking device' get its information from determining that the device is non-responsive from the expiration of a time period. In each reference cited by the examiner, the mimicking device knows that a device will not respond and merely mimics the device upon receipt of a transmission intended for the device. In is completely unnecessary to wait on the expiration of a time period.

Blackborow et al describes a hot connection for a disk drive wherein stand-in SCSI interface **30** present an active and "drive not ready" response when disk cartridge **16** is not in place. However, Blackborow et al teaches that stand-in SCSI interface **30** merely responds to a request for access to disk cartridge **16** because cartridge in/out controller **40** alerts stand-in SCSI interface **30** to the absence of disk cartridge **16**, see col. 6, lines 7 - 19.

Blackborow et al, further describes the functionality of cartridge in/out controller **40** with respect to stand-in SCSI interface **30**:

> The latch 40 provides an output port for the microcontroller 32. There are bit positions to turn on +5 V and +12 V power to the disk cartridge 16, to actuate the load/unload motor 43 to move the cartridge in and out, and a

bit to operate the status LED. All bits are active high. The address for this port is A14 true and is asserted when A14 is true and -WR is also true.

The circuit 48 (FIG. 2A) provides a power on reset controller for the circuit board 10. It resets the interface circuit 30, the microcontroller 32 and the output port latch 40. The resetting of the output port latch 40 insures that all of its control signals are off at power up. The reset circuit 48 also monitors the +5 V and +12 V power buses.
(Blackborow et al. col. 7, lines 52 - 65)

Nowhere does Blackborow et al teach or suggest "**detecting a signal on the bus indicating a request to access the device**" as required by the claims. Furthermore, Blackborow et al does not teach or suggest "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the claims. However, in contrast to the examiner's assertions, it would not be obvious to use a time out period because under Blackborow et al's teaching, the mimicked response is immediate. Blackborow et al teaches away from utilizing a timeout period by the mimicking device (stand-in SCSI interface **30**) acquiring advance information that the device will be non-responsive is called.

Emerson et al is similarly deficient. Emerson et al discloses a computer system that provides for transparent plugging and unplugging of a keyboard independent of system operation. Emerson et al describes the plug-and-play environment discussed above including configuring a virtual keyboard during the POST self test if a keyboard is not present. In this case the mimic device is virtual keyboard **204**. However, rather than virtual keyboard **204** "**detecting a signal on the bus indicating a request to access the device**" and "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the claims, IRC 170 provides virtual keyboard **204** when primary keyboard **157** or secondary keyboard **159** are not available.

Therefore, IRC 170 provides a virtual keyboard 204. Referring ahead to FIGS. 4A-C, it is seen that this virtual keyboard 204 is coupled to the 8042 keyboard controller 156 when no keyboard is present. This allows a standard boot without an attached keyboard 157 or 159. When the CPU executes its POST code, it inquires through the 8042 keyboard controller 156 whether a keyboard 157 or 159 is present. If not, as illustrated in

FIGS. 2C and 4A, the virtual keyboard 204 responds to the 8042 keyboard controller 156, thus making the POST code believe a physical keyboard is present, allowing the system to boot. This virtual keyboard 204 is implemented much as would be the communications logic within a standard keyboard, using the same serial communication structure. The virtual keyboard 204 is further discussed in detail below in conjunction with FIGS. 21-26.
(Emerson et al col. 8, line 7 - 22)

In other words, Emerson et al teaches that the selection of virtual keyboard **204** is a system management interrupt (SMI) function and not a function of the mimic device. Virtual keyboard **204** is given advance knowledge of the missing primary keyboard **157** or secondary keyboard **159**, see col. 7, lines 11 - 35.

This sequence of events is further reflected in FIGS. 4A through 4C, simplified block diagrams illustrating a sequence of operations in which the primary keyboard 157 is connected. In FIG. 4A, no keyboard is connected. Assume that the computer system S has just been powered on. During the POST and OS load routines, standard computers check to ensure that a keyboard is actually connected by writing appropriate commands to the keyboard through the 8042 keyboard controller and examining the resulting signals returned by the keyboard. In this case, no keyboard is connected. Instead, the SMM 212 software has configured the IRC 170 to connect the virtual keyboard 204 to the 8042 keyboard controller 156. Therefore, when the POST or OS load I/O commands are written through the 8042 keyboard controller 156, they are trapped by the virtual keyboard 204. This in turn forces system management interrupt, or SMI, which causes the CPU 50 to enter SMM 212. As is further discussed below in the flowcharts of FIGS. 32A-D, these codes sent to the virtual keyboard 204 are then parsed by the firmware in SMM 212 software, and an appropriate response is sent to the virtual keyboard 204. The virtual keyboard 204 then sends the appropriate responsive scan codes to the 8042 keyboard controller 156, just as an actual keyboard would. The POST and OS load codes, satisfied that a keyboard is present, then continues normal operation.
(Emerson et al col. 13, line 46 to col. 14, line 4)

As to the hot plug operation, Emerson et al teaches that virtual keyboard **204** is invoked whenever the system detects a changes in the current flowing through the primary keyboard **157** or the corresponding secondary keyboard **159** power.

In this way, the presence or absence of a keyboard is detected through minute changes in the current flowing through the primary keyboard 157 supply voltage KBD0VDD or the corresponding secondary keyboard 159 supply voltage KBD1VDD. This presence is indicated by the signals KBDPR0 and KBDPR1, active high signals which internal to the IRC 170 are mapped to two signals KBD.sub.-- PRESEN.sub.-- RAW›1:0!, which are active high signals.
(Emerson et al col. 15, line 29 to col. 17, line 44, col. 17, lines 41 - 49 reproduced) )

Nowhere does Emerson et al teach or suggest "**detecting a signal on the bus indicating a request to access the device**" as required by the claims. Furthermore, Emerson et al does not teach or suggest "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the claims. Instead, Emerson et al teaches to provide the system management interrupt with alternative and virtual devices (keyboards) that can be configured in the event that a primary device not available during the POST self test. Thereafter the system monitors a device's power for fluctuation that indicate that the device is unplugged. Here again there is no need to use a time out period because the mimicking device is given advance information that it will be needed. The mimic device then substitutes the device rather than mimicking it if it fails to respond during a time period. Emerson et al teaches away from utilizing a timeout period by the mimicking device (virtual keyboard 204) acquiring advance information that the request device will be non-responsive when called.

Hartung et al discloses an interface apparatus or Nest fits wherein IDE devices, such as magnetic tape drives and hard drives, are hot swapable in and out of the Nest while the IDE bus is active without degrading the IDE bus. Hartung et al describes a hot swapable nest that relies on a connection from connector 42 for determining device 40 is in place. Hartung et al nether teaches nor suggests "**detecting a signal on the bus indicating a request to access the device**" and "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the claims, but instead uses a signal from connector 42 for alerting Nest 30

that a drive is not available. Nest **30** knows in advance of any request for a device whether or not that device is available from a signal generated by connector **42**.

> Nest 30 is capable of detecting the presence or absence of a device such as device 40. Such detection can be accomplished in many ways, such as sensing of an appropriate signal on a selected terminal upon connection of connector 42, for example. If device 40 is present at reset, then Nest 30 passes through on bus 94 all data and control signals to device 40 and device 40 functions normally as if it were connected directly to ATA/ATAPI peripheral bus 60. If there is no device detected in the Nest 30, then Nest 30 itself responds in a legal fashion through its own set of command and status registers (phantom task file 124). If there is no device in Nest 30, then Nest bus 94 is electrically isolated from ATA/ATAPI peripheral bus 60, thereby preventing spurious signals from the open connector from corrupting ATA/ATAPI peripheral bus 60. Nest 30 maintains the bus isolation until a device has been inserted into bay 32 and until the device has completed its own power-on/reset initialization.
>
> Nest 30 is configured as master, slave, or cable select in the standard method prescribed in the ATA standard. Nest 30 passes the select signal through to the device connector 42 via the cable select line. All devices to be inserted into Nest 30 are preferably configured for cable select. In this manner the inserted device always inherits the master/slave setting of Nest 30 without having to be individually configured.
> (Hartung et al. col. 5, lines 39 to col. 6, line 5)

Nowhere does Hartung et al teach or suggest "**detecting a signal on the bus indicating a request to access the device**" as required by the claims. Furthermore, Hartung et al does not teach or suggest "**sending a response to the signal when a selected period of time passes without a response being made by the device**" as required by the claims. Instead, Hartung et al teaches to provide the Nest with advance information concerning the availability of drives. Prior to any request for access to a drive being received by Nest **30**, it is known whether the drive is available or not. Once again there is no need to use a time out period because the mimicking device is given advance information that it will be needed. The mimic device then responds directly to any request rather than waiting for the expiration of a time period. Hartung et al actually teaches away from utilizing a timeout period by the mimicking device by acquiring

advance information that the requested device (drive) will be non-responsive when called.

Therefore, the rejection of claims 1-40 under 35 U.S.C. § 103 has been overcome.

## IV. Conclusion

It is respectfully urged that the subject application is patentable over any combination of Beckman, Win95, Blackborow et al, Emerson et al and Hartung et al and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 12 August 2002

Respectfully submitted,

Peter P. Scott

Peter P. Scott
*Registration No. 33,279*
**LSI LOGIC CORPORATION**
1551 McCarthy Blvd.
M/S AD-106
Milpitas, CA 95035
719.533.7969 (Voice)
719.533.7254 (Fax)
ATTORNEY FOR APPLICANT

## APPENDIX OF REDACTED MATTERS

**IN THE SPECIFICATION:**

Please amend the paragraph on page 2, line 10, to page 3, line 6, as follows:

A good server data processing system is dependent on an efficient input/output (I/O) system. After all functional requirements have been defined and all the components specified, a server relies heavily on its storage I/O subsystem. This critical subsystem controls the movement of data between the CPU and peripheral devices. No matter how fast the CPU, a single I/O bottleneck can keep any server from living up to its full performance potential. Therefore, an important aspect of building a better server is building an efficient storage subsystem. Today's servers place an extra demand on storage systems. Faster CPUs require more and more data from the storage system to meet increased data accesses. Multitasking operating systems allow data to [for] be processed by multiple threads. Multimedia applications also heavily tax a server's data transfer ability with huge files. Faced with these facts, a server can no longer perform to its optimum unless built with an efficient I/O system. The American National Standards Institute (ANSI) has a standard for defining a standard high-speed parallel interface that is used to provide for the needed speed in I/O transfers. The standard is referred to as the Small Computer System Interface (SCSI) standard. A SCSI interface is used to connect microcomputers to SCSI peripheral devices, such as many hard disks and printers, and to other computers and local area networks. SCSI provides for a local I/O bus that can be operated over a wide range of data rates. The primary objective of the SCSI interface is to provide host data processing systems, such as servers, with device independence within the class of devices. Thus, different disk drives, tape drives, printers, optical media drives, and other devices [and the] can be added to a computer without requiring modifications to system hardware or software.

**IN THE CLAIMS:**

Please amend claims 5, 8, 10, 14, 18, 27, and 35 as follows:

5. (amended) The method of claim 3, wherein the step of sending a response further includes sending a second signal [to] in response to the request.

8. (amended) The method of claim 1 further comprising:

detecting a signal on the bus indicating a request to access a second device;

monitoring the bus for a response by second device; and

sending a response to the signal after a selected period of time passes without a response being made by the second device.

10. (amended) The method of claim 1, wherein the device is connected to the bus and unable to [response] respond to the request within the selected period of time.

14. (amended) A method for emulating a device during initialization of an operating system, wherein the device is configured for use within a data processing system and may be attached to a bus within the data processing system, the method comprising:

monitoring the bus for a signal selecting the device for an input/output transaction during initialization of the operating system;

monitoring the bus for a response by the device in response to detecting the signal selecting the device; and

sending a response to the signal after a selected period of time passes without a response being made by the device, wherein the response indicates to the operating system that the device is present within the data processing system.

18. (amended) A data processing system comprising:

a bus;

detection means for detecting a signal on the bus indicating a request to access [the] a device;

monitoring means for monitoring the bus for a response by the device; and

transmission means for sending a response to the signal after a selected period of time passes without a response being made by the device.

27. (amended) A data processing system comprising:

a bus;

a plurality of devices connected to the bus; and

a mimic device connected to the bus, wherein the mimic device monitors the bus for a signal selecting a selected device within the plurality of devices for an input/output transaction during initialization of an operating system within the data processing system, monitors the bus for a response by the selected device in response to detecting the signal selecting the device, and sends a response to the signal a selected period of time passes without a response being made by the selected device, wherein the response indicates to the operating system that the selected device is present within the data processing system.

35. (amended) A computer program product for use with a data processing system for mimicking a device, a computer program product comprising:

a computer usable medium;

first instructions for detecting a signal on the bus indicating a request to access a device;

second instructions for monitoring the bus for a response by the device; and

third instructions for sending a response to the signal after a selected period of time passes without a response being made by the device, wherein the instructions are embodied within the computer usable medium.

Please add the following new claims:

--41. A method in a data processing system for mimicking a device for use within the data processing system, wherein the device may be connected to a bus, the method comprising:

detecting an input/output (I/O) signal on the bus indicating a request to access the device;

ascertaining that the device being requested is to be mimicked;

monitoring the bus for a response by the device; and

mimicking the device by sending a response to the signal when a selected period of time passes without a response being made by the device.

42. The method of claim 41, wherein the response includes pre-stored data according to a bus protocol.

43. The method of claim 1, wherein ascertaining that the device being requested is to be mimicked; further comprises starting a timer.

44. The method of claim 41, wherein the input/output (I/O) signal is a first input/output (I/O), the device is a first device and the response is a first response, the method further comprises:

detecting a second input/output (I/O) signal on the bus indicating a request to access a second device;

ascertaining that the second device being requested is to be mimicked;

monitoring the bus for a seconds response by the device; and

mimicking the second device by sending a second response to the signal when a selected period of time passes without a second response being made by the second device.

45. The method of claim 41 further comprises:

ascertaining that no further transacting is necessary; and

releasing the bus.--